

# TECHNICAL RESEARCH REPORT

Stability of Wireless Networks for Mode S Radar

by Jay P. Chawla, Steven I. Marcus, Mark A. Shayman

T.R. 2000-43



*ISR develops, applies and teaches advanced methodologies of design and analysis to solve complex, hierarchical, heterogeneous and dynamic problems of engineering technology and systems for industry and government.*

*ISR is a permanent institute of the University of Maryland, within the Glenn L. Martin Institute of Technology/A. James Clark School of Engineering. It is a National Science Foundation Engineering Research Center.*

**Web site** <http://www.isr.umd.edu>

<b>Report Documentation Page</b>			Form Approved OMB No. 0704-0188	
<p>Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p>				
1. REPORT DATE <b>2000</b>	2. REPORT TYPE	3. DATES COVERED <b>00-00-2000 to 00-00-2000</b>		
4. TITLE AND SUBTITLE <b>Stability of Wireless Networks for Model S Radar</b>		5a. CONTRACT NUMBER		
		5b. GRANT NUMBER		
		5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S)		5d. PROJECT NUMBER		
		5e. TASK NUMBER		
		5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) <b>University of Maryland, Electrical Engineering Department, Institute for Systems Research, College Park, MD, 20742</b>		8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)		
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION/AVAILABILITY STATEMENT <b>Approved for public release; distribution unlimited</b>				
13. SUPPLEMENTARY NOTES				
14. ABSTRACT <b>see report</b>				
15. SUBJECT TERMS				
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES <b>24</b>
a. REPORT <b>unclassified</b>	b. ABSTRACT <b>unclassified</b>	c. THIS PAGE <b>unclassified</b>		19a. NAME OF RESPONSIBLE PERSON

# **Stability of Wireless Networks for Mode S Radar<sup>1</sup>**

Jay P. Chawla, Steven I. Marcus, and Mark A. Shayman

Electrical Engineering Department and

Institute for Systems Research

University of Maryland

College Park, MD 20742

Email: chawla@eng.umd.edu

## **Abstract:**

Stability issues in a connectionless, one hop queueing system featuring servers with overlapping service regions (e.g. a Mode Select (Mode S) Radar communications network or part of an Aeronautical Telecommunications Network (ATN) network) are considered, and a stabilizing policy is determined in closed-loop form. The cases of queues at the sources (aircraft) and queues at the servers (base stations) are considered separately. Stabilizability of the system with exponential service times and Poisson arrival rates is equivalent to the solvability of a linear program and if the system is stabilizable, a stabilizing open loop routing policy can be expressed in terms of the coefficients of the solution to the linear program. We solve the linear program for the case of a single class of packets.

## **Introduction:**

Many queuing problems in the literature have addressed the problem of routing messages from a single source to multiple servers or serving multiple queues with a single server. This simple  $1 : n$  or  $n : 1$  topology will not hold in all one-hop queueing networks.

The topology we analyze is based on the Mode S system for the air/ground segment of communication between aircraft and air traffic control. Our system consists of multiple servers with overlapping service areas. This topology also is appropriate for a connectionless satellite or cellular network, or a network in which load sharing is employed as in the Mobile IP standard.

In the Mode Select (Mode S) radar beacon system [2], Mode S base stations communicate with aircraft by “interrogation”. In an interrogation, the base station sends packets to the aircraft. Mode S regions overlap by necessity, and the overlap is largest at high altitudes. For Mode S stations under different controllers, there is a hard boundary between their respective areas of responsibility: aircraft on one side of the region are connected to one base station, and aircraft on the other side are connected to the other. For two base stations under the same controller, control is decided on a per-aircraft basis for aircraft in the overlap of their coverage regions. This decision is made by the central ground controller, and uses dynamic information. The location of the aircraft is never in question, nor must the location be measured by signal strength; the Mode S stations

---

<sup>1</sup>This paper is based on [14], written by the same authors

are also responsible for radar tracking of the aircraft, and the central ground station has a composite picture of all the aircraft in the area. The Mode S radar beacon system is only a part of the Aeronautical Telecommunications Network (ATN), which is a network of networks comprising Mode S, Satellite, and VHF Data Link for the air/ground link. The results in this paper apply a fortiori to the air/ground link of the ATN.

In this paper, we use the equivalence of stabilizability (defined below) of queueing problems involving routing decisions to the existence of a solution to a linear program. The linear program can be explicitly solved for the case of a single class of packets, and we determine a stabilizing open loop control thereby.

Linear programming techniques have been used ([3], [4], [8], [9], [10], [12], and [13]) to analyze stability of queueing networks. In [3], [4], [10], and [12], linear programming techniques were used to determine stabilizability of scheduling problems on reentrant lines. In [8], [9], and [13], linear programming techniques were applied to routing problems.

Kumar, Down, and Meyn have explored the use of quadratic and piecewise linear cost functions in linear programs to demonstrate stability of a queueing network under a specific policy and also under a class of policies. The method to determine stability of a policy is to construct a Lyapunov function using quadratic or piecewise linear cost functions and then to use Foster's criterion. LPs are constructed, which, if solved, would prove stabilizability. In [4], Kumar showed the equivalence of the existence of a solution to a linear program (LP) to the stabilizability of a scheduling problem. In [12], Kumar further found linear programs which, if solvable, implied the stability of all non-idling policies for some scheduling problems. In [10], Down and Meyn analyzed stability of reentrant line queueing systems. In [3], Kumar and Meyn used linear and nonlinear programs to deduce an appropriate quadratic Lyapunov function.

In this paper, we tie the stabilizability of a routing problem to the solvability of a linear program.

In [9], Tassiulas and Ephremides found an equivalence between stabilizability of a general queueing network and the solvability of a linear program in constraint form. Tassiulas and Ephremides also examine a multi-hop queueing system in [8], deriving stabilizability conditions and demonstrating a closed-loop stabilizing policy that is a special case of the policy constructed in [9]. Also, in [13], Tassiulas generalizes the results to systems with a time-varying topology. Our stabilizability result is a special case of the multi-hop result in [9], but here we solve the linear program explicitly in the case of a single class of packets, yielding a stabilizing open loop policy. Similarly, if the linear program of [9] is solved, a stabilizing policy for the more general, multi-hop network is given in terms of the solution to the linear program.

We are concerned with controlled Markov chains describing queueing systems. We define a Markov chain consisting of the vector of queue lengths in a queueing network to be stable if it converges to an ergodic distribution with finite average queue length in each queue, and we define a controlled queueing network or system to be stabilizable if there exists a stationary control policy that induces a stable Markov chain.

The contributions of this paper are the application of linear programming techniques

to a routing problem and the determination of open loop stabilizing policies for that problem by solving the linear program in the case of a single class of packets. Open loop control policies are of particular interest because in a real network, it is often difficult for a controller at one node to obtain queue length information at other nodes.

We make use of an equivalence between discrete time and continuous time systems called “uniformization”. This equivalence was discussed by Serfozo in [5] (see also references therein), and appears in the context of a discounted optimal cost problem in [7] (page 283).

The paper is organized as follows: In Section 1, the problem is described and open loop stabilizability results are presented which show that the solvability of a linear program is a necessary and sufficient condition for stabilizability of the system, and a stabilizing open loop policy is determined in terms of the coefficients of the solution to the linear program. Sections 2 and 3 contain the main results of this paper: explicit solutions of the linear program for two cases. The proofs of the results in Sections 2 and 3 are presented in the appendices. The proofs of Propositions 1 and 2 in Section [1] are omitted; they can be deduced from the general stability result in [9].

### **Section 1: The general shared-queues problem**

Let there be  $n$  servers and  $m = C \cdot (2^n - 1)$  arrival processes. Label the servers  $\{S_1, \dots, S_n\}$ . Each arrival process is dedicated to a subset  $L \subset \{1, \dots, n\}$  of the servers. Also, each arrival process contains only class  $j$  packets, where  $j = 1, \dots, C$ . Since there are  $2^n - 1$  nonempty subsets of  $\{1, \dots, n\}$  and  $C$  classes of packets, there are  $m$  arrival processes. Label the arrival processes  $\{A_L^j\}$ .<sup>1</sup> We will examine two systems with this configuration of servers and arrival processes: the controlled arrival process problem (Figure 0.2) and the controlled service process problem (Figure 0.1). (Note: the figures depict the problems for the simple case of 2 servers and 1 class of packets.) In the controlled arrival process problem, there are  $n$  queues, 1 dedicated to each server. Label these queues  $\{q_1, \dots, q_n\}$ . At the time a packet arrives in arrival process  $A_L^j$ , the controller must route it to a queue  $q_i$ ;  $i \in L$ . In the controlled service process problem, there are  $m$  queues, one for each arrival process. Label the queue to which packets from arrival process  $A_L^j$  are routed  $q_L^j$ . In this setup, the servers must select a nonempty queue to serve whenever they are idle. Label the queue lengths  $x_i$ , where  $i \in \{1, \dots, n\}$  in the controlled arrival process problem and  $x_L^j$ ,  $L \subset \{1, \dots, n\}$ ,  $j \in \{1, \dots, C\}$  in the controlled service process problem. A queue can be served by any of the servers which share it (packet class of a queue does not affect which servers may serve the queue) in the controlled service process problem, and only by its server in the controlled arrival process problem. In the controlled arrival process problem, packets in each of the  $n$  dedicated queues are served on a FIFO basis. In the controlled service process problem, if  $k$  packets are in a shared queue, no more than  $k$  servers may serve it at one time. Let the set of queues be called  $Q$  and call the set of queue lengths  $X$ . Each server has a constant service rate  $\mu_i$  and each packet contains an amount

---

<sup>1</sup>Theorems 1 and 2 concern systems with a single class of packets, so we will drop the ‘ $j$ ’ from the notation in those sections.

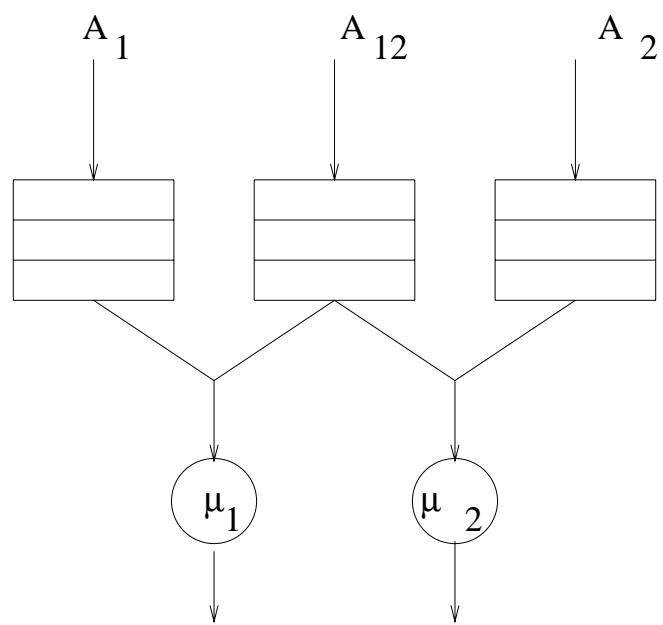


Figure 0.1: Controlled Service Process: Queues at the Arrival Streams

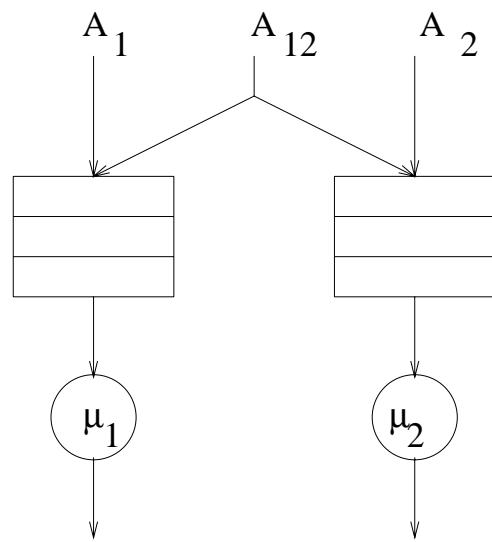


Figure 0.2: Controlled Arrival Process: Queues at the Servers

of work which is i.i.d. exponentially distributed with parameter 1, and arrival process  $A_L^j$  undergoes Poisson arrivals at rate  $\lambda_L^j$ . The arrival times and the work initially contained in the packets are all mutually independent. Also, preemption is not permitted: once a server begins serving a packet, it must complete service on that packet before serving another packet.

**Remark:** While the controlled arrival process problem is a routing problem, the controlled service process problem contains elements of both routing and scheduling since the controller must decide not only which queue to serve with an idle server, but the order in which to serve the queues.

**Remark:** The model of constant service rates and exponential work contained in a packet is equivalent in distribution to a model with exponential service times.

### The controlled service process problem

The system introduced in [1] can be transformed into the controlled service process problem with one class of packets. In [1], a necessary and sufficient condition for stabilizability of the single-class ( $C = 1$ ) system was found and a closed-loop policy, “serve the shortest queue” was demonstrated to stabilize the system when it is stabilizable. The stabilizability condition for  $C = 1$  is:

$$\forall K \subset \{1, \dots, n\}, \sum_{i \in K} \mu_i > \sum_{L \subset K} \lambda_L \quad (1.1)$$

The intuition behind (1.1) is that for every set of servers ( $\{S_i | i \in K\}$ ), the total service rate of the set is greater than the rate of work arriving that *must* be served by a server from that set.

We now examine the stability of such systems under multiple classes of arrivals. Suppose that the shared queue  $q_L$  undergoes Poisson arrivals of  $C$  different classes, with the arrival rate of class  $c_j$  packets equal to  $\lambda_L^j$ . Suppose that the service rate of server  $S_i$  now depends on both the queue and the class of packet being served, such that the service time for server  $S_i$  serving a class  $c_j$  packet in queue  $q_L$  is exponential with parameter  $\mu_i^{j,L}$ . Also, consider a packet in service as not being in queue. The question is: *Under what conditions (parameter sets) is this system stabilizable?* (We do not allow preemption of packets in service.)

In an open loop policy, it is possible for a server to elect to serve a queue which is empty. When that happens, we say that the server is serving a “null” packet, and the latency time for the server to complete service of the null packet has the same distribution as if it were serving a real packet.

We claim that the system is open loop stabilizable iff it is stabilizable, and we define a set of open loop Bernoulli service allocations as follows: When server  $S_i$  completes service on a packet (or a null packet if it was serving a class of packet in a queue containing no packets of that class), then it selects a class  $j$  packet from queue  $q_L$  with probability

$p_i^{j,L}$ . If queue  $L$  contains a packet of class  $j$ , then the packet is removed immediately from queue and processed in time distributed exponentially with parameter  $\mu_i^{j,L}$ . If there is no such packet in queue, server  $S_i$  is idle for the same period of time.

**Proposition 1:**

The system with controlled service process is stabilizable if and only if the following linear program has a solution with  $\theta < 0$ . Furthermore, if  $\theta < 0$ , then the system is stabilized by the set of open loop Bernoulli service allocations

$$p_i^{j,L} \doteq \frac{z_i^{j,L} \cdot \mu_i^{j,L}}{\sum_{j=1,\dots,C; L \ni i} z_i^{j,L} \cdot \mu_i^{j,L}}$$

LP:  $\min \theta$  such that:

$$z_i^{j,L} \geq 0; i = 1, \dots, n; j = 1, \dots, C; L \ni i$$

$$\sum_{j=1,\dots,C; L \ni i} z_i^{j,L} = 1; i = 1, \dots, n$$

$$\lambda_L^j - \sum_{i \in L} \mu_i^{j,L} z_i^{j,L} \leq \theta; j = 1, \dots, C; L \subset \{1, \dots, n\}$$

The solution to the LP is  $\{\theta, \{z_i^{j,L}\}\}$ .

**Proof:**

The proof is omitted as this proposition can be shown to be a special case of the theorem proved by Tassiulas and Ephremides in [9].

The intuition behind the LP is as follows:  $z_i^{j,L}$  is the fraction of the time that server  $S_i$  spends serving a class  $j$  packet in queue  $q_L$  in the limit (the ergodic distribution). The first constraint ensures that each server is not utilized more than all of the time, while the second constraint ensures (if  $\theta < 0$ ) that the arrival rate of class  $j$  packets to queue  $q_L$  is less than the total ergodic service rate (the product of fraction of time spent and exponential service time parameter) applied to class  $j$  packets in queue  $q_L$  from all servers. The binomial splitting probabilities are found by normalizing the fraction of time spent ( $z$ ) by the exponential service time parameter ( $\mu$ ).

**The controlled arrival process problem**

It can be shown using the general stabilizability result from [8] that the condition for stabilizability of the controlled arrival process problem with one class of packets is given by (1.1). Thus it is the same as the stabilizability condition for the controlled service process problem with a single class of packets.

The simplicity of the stability condition for the case of a single class of packets indicates that a simple solution may exist for the linear program in that case. In the proofs of Theorems 1 and 2, the linear programs of Propositions 1 and 2 are solved iteratively by repeatedly reducing the dimensionality of the system until it consists of a set of  $n$  stable  $M/M/1$  queues.

In the multi-class case, the stability condition must be different for the controlled arrival process problem and the controlled service problem since the parameters are different. Specifically, in the controlled service process problem, the service rate  $\mu_i^{j,L}$  depends on three parameters, whereas in the controlled arrival process problem, the service rate  $\mu_i^j$  depends on only two parameters. But as remarked after Proposition 2, when  $\mu_i^{j,L}$  is independent of  $L$ , the stability conditions are the same.

Define the set of open loop Bernoulli routing probabilities  $\{p_i^{j,L}\}$  as follows: When a packet of class  $j$  arrives in arrival process  $A_L$ , we assign it to queue  $q_i$ ;  $i \in L$  with probability  $p_i^{j,L}$ .

### **Proposition 2:**

The system with controlled arrival process is stabilizable if and only if the following linear program has a solution with  $\theta < 0$ . Furthermore, if  $\theta < 0$ , then the system is stabilized by the set of open loop Bernoulli routing probabilities  $\{p_i^{j,L}\}$ .

LP:  $\min \theta$  such that:

$$p_i^{j,L} \geq 0; i = 1, \dots, n; j = 1, \dots, C; L \ni i$$

$$\sum_{i \in L} p_i^{j,L} = 1; L \subset \{1, \dots, n\}, j = 1, \dots, C$$

$$\sum_{j=1}^C \frac{\sum_{L \ni i} p_i^{j,L} \lambda_L^j}{\mu_i^j} < 1 + \theta; i = 1, \dots, n$$

The solution to the LP is  $\{\theta, \{p_i^{j,L}\}\}$ .

### **Proof:**

The proof is omitted; this proposition is a special case of a theorem proved by Tassiulas and Ephremides in [9].

The intuition behind the LP is as follows:  $p_i^{j,L}$  is the fraction of packets from arrival process  $A_L^j$  that are routed to server  $S_i$ . The first constraint ensures that the splitting probabilities of packets in arrival stream  $A_L^j$  sum to 1, while the second constraint ensures that server  $S_i$ 's utilization ( $\sum_{j=1}^C \frac{\sum_{L \ni i} p_i^{j,L} \lambda_L^j}{\mu_i^j}$ ) is less than  $1 + \theta < 1$  if  $\theta < 0$ .

**Remark:** This LP is solvable with  $\theta < 0$  iff the LP of Proposition 1 is solvable with  $\theta < 0$  for  $\mu_i^{j,L} = \mu_i^j \forall L \ni i$ , i.e. the controlled arrival process problem and the controlled

service process problem are stabilizable under the same parameter sets if the service time distribution in the controlled service process problem is independent of  $L$ , the queue served, and is identical to the service time in the controlled arrival process problem (where there is only one queue accessible to each server).

## Section 2: Open-loop policies for the controlled arrival process problem with a single class of packets

The stability result in [8] implies the stability of the “route to the shortest queue” policy for the case of a single class of packets ( $C = 1$ ) when the system is stabilizable. An open loop policy that stabilizes the system can be determined by applying the flow rates of the stationary distribution induced by any stabilizing policy. (A flow rate of a stationary distribution is the expected fraction of the time an arrival to a given arrival process is routed to a given queue.) The intuition behind this is that, since the system is ergodic and irreducible under the “route to the shortest queue” policy, the limit as  $t$  goes to infinity of the rate at which any given arrival process routes its packets to each of the queues to which it is connected exists. We know that the sum of the rates at which each arrival process routes its packets to a queue must be less than the service rate of that queue because the system is stable under the “route to the shortest queue” policy.

Therefore, if we use open loop Bernoulli routing at the arrival processes, splitting at the rate given by this limit, then we will obtain stability as the arrival rate of packets to each queue will be less than the service rate. The problem with this approach is that computation of these rates requires knowledge of the limiting distribution. However, we may construct a stable open loop Bernoulli routing policy given knowledge only of the arrival and service rates by using the following algorithm, as shown in the proof of Theorem 1.

### Algorithm 1

GIVEN: The arrival rates  $\{\lambda_L | L \subset \{1, \dots, n\}\}$  and the service rates  $\{\mu_i | i \in \{1, \dots, n\}\}$

CONSTRUCT: a stabilizing set of Bernoulli routing probabilities  $\{p_i^L | i \in L \subset \{1, \dots, n\}\}$

Do(j=n,n-1,...,2)

$\forall L \subset \{1, \dots, n\}$  such that  $|L| = j$

Select  $B, D \subset L$  such that  $B \cap D = \emptyset$  and  $B \cup D = L$

Select two numbers<sup>1</sup>  $b$  and  $d$  such that  $b + d = \lambda_L$  and the system with the following

<sup>1</sup>The constraints on  $b$  and  $d$  are given in terms of  $\Psi^K$ , defined as follows:

$$\Psi^K = \sum_{S_i \in S^K} \mu_i - \sum_{A_L \in A^K} \lambda_L \quad (2.1)$$

$\Psi^K$  is the “slack” available between service and arrival rates for subset  $K$ , i.e. between the servers  $S^K \doteq \{S_i; i \in K\}$  and the arrival processes  $A^K \doteq \{A_L; L \subset K\}$ . The system is stabilizable iff  $\Psi^K > 0 \ \forall K$ .

modified arrival rates is still stable:

$\lambda_L = 0$ ;  $\lambda_B$  is increased by  $b$ ;  $\lambda_D$  is increased by  $d$ ; all other arrival rates unchanged.

Then modify the system and define the Bernoulli routing probabilities  $\{p_i^L | i \in L\}$  as follows:

$$p_i^L = \frac{b}{\lambda_L} p_i^B + \frac{d}{\lambda_L} p_i^D$$

Then set  $p_i^i = 1 \forall i$  and the recursion is complete.

### Theorem 1

The queuing system of the ‘controlled arrival process problem’ with one class of packets is stabilizable iff (1.1) holds.

Furthermore, if the system is stabilizable, there exists an open loop stabilizing policy which can be computed explicitly through the recursive procedure defined in Algorithm 1.

Proof: see Appendix 1

### Example 1

Now let us solve an example problem (i.e. recursively determine the open loop Bernoulli routing probabilities.) Consider the arrival processs problem with three servers such that  $\mu_i = 1; i = 1, 2, 3$  and where the arrival rates for the seven arrival processes are:

$$\lambda_1 = .2; \lambda_2 = .3; \lambda_3 = .1; \lambda_{1,2} = .4; \lambda_{1,3} = .5; \lambda_{2,3} = .6; \lambda_{1,2,3} = .6$$

The following procedure is Algorithm 1 applied to the above system.

First, let us break arrival process  $\lambda_{1,2,3}$  into two pieces, say  $\{1, 2\}$  and  $\{3\}$ . We see that  $\Psi^{1,2} = \mu_1 + \mu_2 - \lambda_1 - \lambda_2 - \lambda_{1,2} = 1.1$ . Similarly,  $\Psi^3 = \mu_3 - \lambda_3 = .9$ . Because  $\lambda_{1,2,3} = .6 < \Psi^3$ , we can set  $p_3^{1,2,3} = 1$ , giving us the following new set of arrival rates:

$$\lambda_1 = .2; \lambda_2 = .3; \lambda_3 = .1 + .6 = .7; \lambda_{1,2} = .4; \lambda_{1,3} = .5; \lambda_{2,3} = .6; \lambda_{1,2,3} = 0$$

Now, let us split arrival process  $\lambda_{2,3}$ . We see that  $\Psi^2 = .7$  and  $\Psi^3 = .3$ . Because  $\lambda_{2,3} = .6 < \Psi^2$ , we can set  $p_2^{2,3} = 1$ , giving us the following new set of arrival rates:

$$\lambda_1 = .2; \lambda_2 = .3 + .6 = .9; \lambda_3 = .7; \lambda_{1,2} = .4; \lambda_{1,3} = .5; \lambda_{2,3} = 0; \lambda_{1,2,3} = 0$$

Now, let us split arrival process  $\lambda_{1,2}$ . We see that  $\Psi^1 = .8$  and  $\Psi^2 = .1$ . Because  $\lambda_{1,2} = .4 < \Psi^1$ , we can set  $p_1^{1,2} = 1$ , giving us the following new set of arrival rates:

---

The stability constraints on  $b$  and  $d$  are that  $b < \Psi^K \forall K$  such that  $B \subset K, D \not\subset K$ , and that  $d < \Psi^K \forall K$  such that  $D \subset K, B \not\subset K$ .

$$\lambda_1 = .2 + .4 = .6; \lambda_2 = .9; \lambda_3 = .7; \lambda_{1,2} = 0; \lambda_{1,3} = .5; \lambda_{2,3} = 0; \lambda_{1,2,3} = 0$$

Now, we split arrival process  $\lambda_{1,3}$ . We see that  $\Psi^1 = .4$  and  $\Psi^3 = .3$ .  $\lambda_{1,3} = .5$  is therefore too large to send to either server exclusively, so let us set  $p_1^{1,3} = \frac{3}{5}$  and  $p_3^{1,3} = \frac{2}{5}$ , giving us the following set of arrival rates:

$$\lambda_1 = .6 + .3 = .9; \lambda_2 = .9; \lambda_3 = .7 + .2 = .9; \lambda_{1,2} = 0; \lambda_{1,3} = 0; \lambda_{2,3} = 0; \lambda_{1,2,3} = 0$$

Thus we have determined a stabilizing, open loop policy.

It is interesting to point out that, since Theorem 1 ensures that the recursive algorithm for determining a set of stabilizing Bernoulli splitting probabilities will work when the system is stabilizable, one can just apply the algorithm in order to determine whether the system is stabilizable. The system is stabilizable iff the algorithm terminates successfully.

### Section 3: Open loop stability of the controlled service process problem with a single class of packets

In addition to assuming there is a single class of packets, we make the assumption that service time depends only on the server, i.e., it is independent of the queue served. The controlled service process problem under these assumptions can be shown to be equivalent to a system first introduced in discrete time by Tassiulas and Ephremides [1]. The continuous-time framework we use here is equivalent to their discrete-time framework by uniformization [5]. In [1], Tassiulas and Ephremides found necessary and sufficient conditions for the system to be stabilizable under the assumptions of this section. In the proof of Theorem 2, we use the same techniques used in the proof of Theorem 1 to explicitly determine an open loop stabilizing policy when a stabilizing policy exists.

#### Theorem 2

The queueing system of the ‘controlled service process problem’ with one class of packets and service time independent of queue served is stabilizable iff (1.1) holds.

Furthermore, if the system is stabilizable, then there exists an open loop stabilizing policy which can be computed explicitly through the recursive procedure specified in Algorithm 2.

Proof: see Appendix 2

Theorem 2 is a special case of Theorem 2a, which is proved in the appendix. Theorem 2a considers a discrete time system, but the system in Theorem 2a can be transformed through uniformization [5] into a continuous time system. The system of Theorem 2a allows servers to serve an arbitrary subset of the set of all queues. Thus, we see that the restriction in Theorem 2, which requires each server to serve only queues that contain its

index, is less general than Theorem 2a. Algorithm 2 will determine a stabilizing open loop Bernoulli service policy for the controlled service process problem iff the system is stabilizable.

In recursively reducing the arrival rates in Algorithm 1, there was no need to use artificial representations because each subset of servers had its own arrival process, so as arrival processes were split, they decomposed into other arrival processes. In Algorithm 2, the situation is different. There are only  $n$  servers, but there are  $2^n - 1$  possible subsets of queues. As these service sets are decomposed, we pass through artificial subsets of queues on the way down to the actual Bernoulli splitting probability to a single queue. Therefore, new notation is required before the presentation of Algorithm 2. Note: Theorem 2a, proved in Appendix 2, is more general in that it permits a server to be dedicated to any subset of queues.

Notation: In the following discussion, capital English letters denote subsets of  $\{1, 2, \dots, n\}$  and capital Greek letters denote sets of subsets of  $\{1, 2, \dots, n\}$ , with the exception that  $B$  and  $D$  also denote sets of subsets of  $\{1, 2, \dots, n\}$ .

Let us rewrite the Bernoulli splitting probability  $p_i^L$  as  $p_{\{\mathcal{K}|i \in K\}}^L$ . Here, we use  $p_\Theta^L$ , where  $L \subset \{1, 2, \dots, n\}$  and  $\Theta \subset \{K|K \subset \{1, 2, \dots, n\}\}$  to represent the Bernoulli probability that a server capable of serving any queue in the set  $\{q_K|K \in \Theta\}$  will serve queue  $q_L$ . Let us extend this further by using the notation  $p_\Theta^\Phi$ , where  $\Theta, \Phi \subset \{K|K \subset \{1, 2, \dots, n\}\}$ ,  $\Phi \subset \Theta$ , to represent the Bernoulli probability that a server capable of serving any queue in the set  $\{q_K|K \in \Theta\}$  will serve a queue selected from the set  $\{q_K|K \in \Phi\}$ . Also, instead of  $\mu_i$ , the service rate of server  $S_i$ , let us write  $\mu_{\{\mathcal{K}|i \in K\}}$ . In general,  $\mu_\Theta$ , where  $\Theta \subset \{K|K \subset \{1, 2, \dots, n\}\}$ , will be used to represent the service rate of the server capable of serving any queue  $q_K|K \in \Theta$ . These changes in notation for the Bernoulli splitting probability and the service rate create a natural representation for use in describing Algorithm 2.

The set  $\{K|i \in K\}$  has cardinality  $2^{n-1}$ , and before the Bernoulli splitting probabilities of each server are determined, we have only that  $p_{\{\mathcal{K}|i \in K\}}^{\{K|i \in K\}} = 1 \forall i \in \{1, 2, \dots, n\}$ . Our objective is to reduce these down to the  $p_{\{\mathcal{K}|i \in K\}}^K$ s, or in the original notation, the  $p_i^K$ s.

### **Algorithm 2**

GIVEN: The arrival rates  $\{\lambda_L|L \subset \{1, \dots, n\}\}$  and the service rates  $\{\mu_{\{\mathcal{K}|i \in K\}}|i \in \{1, \dots, n\}\}$

CONSTRUCT: a stabilizing set of Bernoulli splitting probabilities  $\{p_{\{\mathcal{K}|i \in K\}}^L|i \in L \subset \{1, \dots, n\}\}$

Do(j=2<sup>n-1</sup>,2<sup>n-1</sup>-1,...2)

$\forall \Theta \subset \{K|K \subset \{1, 2, \dots, n\}\}$  such that  $|\Theta| = j$

  if  $\mu_\Theta > 0$ , then

    Select  $B, D \subset \Theta$  such that  $B \cap D = \emptyset$  and  $B \cup D = \Theta$

Select two numbers<sup>1</sup>  $b$  and  $d$  such that  $b + d = \mu_\Theta$  and the system with the following modified service rates is still stable:

$\mu_\Theta = 0$ ;  $\mu_B$  is increased by  $b$ ;  $\mu_D$  is increased by  $d$ ; all other service rates unchanged.

Then set  $p_\Theta^B = \frac{b}{b+d}$ ,  $p_\Theta^D = \frac{d}{b+d}$ .

Solve for each  $p_{\{K|j \in K\}}^L$  by multiplying the appropriate chain of intermediate splitting probabilities together.

### Example 2

The controlled service process problem takes a long time to break down because intermediate sets of queues have to be stepped through, so an example with three servers would be tedious. Let us examine a problem with two servers with service rates  $\mu_1 = 3$  and  $\mu_2 = 2$ . Let the arrival rates to the three queues be given by  $\lambda_1 = 1$ ,  $\lambda_2 = 1$ , and  $\lambda_{\{1,2\}} = 2$ .

Following Algorithm 2, we start out with:

$$\mu_{\{\{1\},\{1,2\}\}} = 3; \mu_{\{\{2\},\{1,2\}\}} = 2$$

Now let us split service process  $\mu_{\{\{1\},\{1,2\}\}}$  into two pieces:  $\mu_{\{\{1\}\}}$  and  $\mu_{\{\{1,2\}\}}$ . This split will not affect the values of  $\Psi^{\{2\}}$ ,  $\Psi^{\{\{1\},\{1,2\}\}}$ , or  $\Psi^{\{\{1\},\{1,2\},\{2\}\}}$ , but it will affect the values of the other  $\Psi$ s. Their current values are:

$$\Psi^{\{\{1\}\}} = \mu_{\{\{1\},\{1,2\}\}} - \lambda_1 = 2$$

$$\Psi^{\{\{1,2\}\}} = \mu_{\{\{1\},\{1,2\}\}} + \mu_{\{\{2\},\{1,2\}\}} - \lambda_{\{1,2\}} = 3$$

$$\Psi^{\{1\},\{2\}} = \mu_{\{\{1\},\{1,2\}\}} + \mu_{\{\{2\},\{1,2\}\}} - \lambda_{\{1\}} - \lambda_{\{2\}} = 3$$

$$\Psi^{\{\{1,2\},\{2\}\}} = \mu_{\{\{1\},\{1,2\}\}} + \mu_{\{\{2\},\{1,2\}\}} - \lambda_{\{1,2\}} - \lambda_{\{2\}} = 2$$

Splitting  $\mu_{\{\{1\},\{1,2\}\}}$  will decrement the value of  $\Psi^{\{\{1\}\}}$  by  $\mu_{\{\{1,2\}\}}$ , will decrement the value of  $\Psi^{\{\{1,2\}\}}$  by  $\mu_{\{\{1\}\}}$ , will decrement the value of  $\Psi^{\{1\},\{2\}}$  by  $\mu_{\{\{1,2\}\}}$ , and will decrement the value of  $\Psi^{\{\{1,2\},\{2\}\}}$  by  $\mu_{\{\{1\}\}}$ .

Therefore, we are constrained that  $\mu_{\{\{1,2\}\}} + \mu_{\{\{1\}\}} = 3$ ,  $\mu_{\{\{1\}\}} < 2$ , and  $\mu_{\{\{1,2\}\}} < 2$ .

We choose  $\mu_{\{\{1\}\}} = 1.5$  and  $\mu_{\{\{1,2\}\}} = 1.5$ , giving us splitting probabilities of  $p_{\{\{1\},\{1,2\}\}}^{\{1\}} = \frac{1.5}{3}$  and  $p_{\{\{1\},\{1,2\}\}}^{\{1,2\}} = \frac{1.5}{3}$ , and service rates of:

$$\mu_{\{\{1\}\}} = 1.5; \mu_{\{\{1,2\}\}} = 1.5; \mu_{\{\{2\},\{1,2\}\}} = 2$$

Now let us split service process  $\mu_{\{\{2\},\{1,2\}\}}$  into two pieces:  $\mu_{\{\{2\}\}}$  and an increment to

---

<sup>1</sup>The constraints on  $b$  and  $d$  are given in terms of  $\Psi^\Phi$ ,  $\Phi \subset \{K|K \subset \{1, 2, \dots, n\}\}$ , defined as follows (Note: this is different from the definition of  $\Psi^K$  for the controlled arrival process problem.):

$$\Psi^\Phi = \sum_{\Omega \subset \{K|K \subset \{1, 2, \dots, n\}\} \mid \Omega \cap \Phi \neq \emptyset} \mu_W - \sum_{K \in \Phi} \lambda_K \quad (3.1)$$

$\Psi^\Phi$  is the “slack” available between service and arrival rates for subset  $\Phi$ , i.e. between the servers with service rates  $\mu_K$ ;  $K \subset \Phi$ , and the arrival processes with rates  $\lambda_K$ ;  $K \in \Phi$ .

The stability constraints on  $b$  and  $d$  are that  $\Psi^\Phi > b \forall \Phi$  such that  $\Phi \cap B \neq \emptyset$  and  $\Phi \cap D = \emptyset$ , and that  $\Psi^\Phi > d \forall \Phi$  such that  $\Phi \cap D \neq \emptyset$  and  $\Phi \cap B = \emptyset$ .

$\mu_{\{1,2\}}$ . This split will not affect the values of  $\Psi^{\{1\}}$ ,  $\Psi^{\{2\},\{1,2\}}$ , or  $\Psi^{\{1\},\{1,2\},\{2\}}$ , but it will affect the values of the other  $\Psi$ s. Their current values are:

$$\Psi^{\{2\}} = \mu_{\{2\},\{1,2\}} - \lambda_2 = 1$$

$$\Psi^{\{1,2\}} = \mu_{\{1,2\}} + \mu_{\{2\},\{1,2\}} - \lambda_{\{1,2\}} = 1.5$$

$$\Psi^{\{1\},\{2\}} = \mu_{\{1\}} + \mu_{\{2\},\{1,2\}} - \lambda_{\{1\}} - \lambda_{\{2\}} = 1.5$$

$$\Psi^{\{1,2\},\{1\}} = \mu_{\{1\}} + \mu_{\{1,2\}} + \mu_{\{2\},\{1,2\}} - \lambda_{\{1,2\}} - \lambda_{\{1\}} = 2$$

Splitting  $\mu_{\{2\},\{1,2\}}$  will decrement the value of  $\Psi^{\{2\}}$  by the increment to  $\mu_{\{1,2\}}$ , will decrement the value of  $\Psi^{\{1,2\}}$  by  $\mu_{\{2\}}$ , will decrement the value of  $\Psi^{\{1\},\{2\}}$  by the increment to  $\mu_{\{1,2\}}$ , and will decrement the value of  $\Psi^{\{1,2\},\{1\}}$  by  $\mu_{\{2\}}$ .

Therefore, we are constrained that  $\mu_{\{2\}} +$  the increment to  $\mu_{\{1,2\}} = 2$ ,  $\mu_{\{2\}} < 1.5$ , and the increment to  $\mu_{\{1,2\}} < 1$ .

We choose  $\mu_{\{2\}} = 1.25$  and the increment to  $\mu_{\{1,2\}} = .75$ , giving us splitting probabilities of  $p_{\{2\},\{1,2\}}^{(2)} = \frac{1.25}{2}$  and  $p_{\{2\},\{1,2\}}^{(1,2)} = \frac{.75}{2}$ , and service rates of:

$$\mu_{\{1\}} = 1.5; \mu_{\{1,2\}} = 2.25; \mu_{\{2\}} = 1.25.$$

And we have found a stabilizing open loop control.

## Conclusion

We have shown closed-form algorithms to quickly solve for an open loop control given that the system is stabilizable. Furthermore, each algorithm can be applied as a check on stabilizability. If the algorithm terminates successfully, the system is stabilizable. If it doesn't terminate successfully, the system is not stabilizable. The two main results proved in Theorems 1 and 2 display an interesting duality, best seen by comparing (2.1) and (3.1).

**Acknowledgement:** The authors would like to thank Leandros Tassiulas for helpful discussions. This research was supported in part by the National Science Foundation under grants EEC-9402384, ECS-9626399, and also under DoD contract Number MDA90497C3015.

## References

- [1] L. Tassiulas and A. Ephremides, "Dynamic Server Allocation to Parallel Queues with Randomly Varying Connectivity", *IEEE Transactions on Information Theory*, vol. 39, no. 2, March 1993, pp. 466-478.
- [2] V.A. Orlando and P.R. Droulihet, "Mode S Beacon System: Functional Description", *MIT Lincoln Labs Project Report ATC-42, revision D*, DOT/FAA/PM-86/19, August 29, 1986.
- [3] P.R. Kumar and S. Meyn, "Stability of Queueing Networks and Scheduling Policies", *IEEE Transactions on Automatic Control*, Vol. 40, No. 2, February 1995, pp. 251-260.

[4] P.R. Kumar, "Scheduling queueing networks: Stability, Performance Analysis and Design", *Stochastic Networks, IMA Volumes in Mathematics and its Applications*, Frank P. Kelly and Ruth William, Editors, Volume 71, Springer-Verlag, New York, 1995, pp. 21-70.

[5] R. F. Serfozo, "An Equivalence Between Continuous and Discrete Time Markov Decision Processes", *Operations Research*, vol. 27, no. 3, May-June 1979, pp. 617-621.

[6] S. Asmussen, *Applied Probability and Queues*, John Wiley & Sons, 1987.

[7] J. Walrand, *An Introduction to Queueing Networks*, Prentice Hall, 1988.

[8] L. Tassiulas and A. Ephremides, "Throughput Properties of a Queueing Network with Distributed Dynamic Routing and Flow Control", *Adv. Appl. Prob.* 28, 1996, pp. 285-307.

[9] L. Tassiulas and A. Ephremides, "Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks", *IEEE Transactions on Automatic Control*, vol. 37 no. 12, December 1992, pp. 1936-1948.

[10] D.G. Down and S. P. Meyn, "A Survey of Markovian Methods for Stability of Networks", *Proceedings of the 11th International Conference on Analysis and Optimization of Systems*, Guy Cohen and Jean-Pierre Quadrat (Eds.), Springer-Verlag 1994.

[11] S.P. Meyn and R.L. Tweedie, *Markov Chains and Stochastic Stability*, Springer Verlag, 1993

[12] P.R. Kumar, "A tutorial on some new methods for performance evaluation of queueing networks", *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, August 1995, pp. 970-980

[13] L. Tassiulas, "Scheduling and Performance Limits of Networks with Constantly Changing Topology", *IEEE Transactions on Information Theory*, vol. 43, no. 3, May 1997, pp. 1067-1073

[14] J. Chawla, S. Marcus, and M. Shayman, "Stability of Wireless Networks for Mode S Radar", *Proceedings of 32nd Conference on Information Sciences and Systems*, Princeton, NJ, March 1998

#### Note on the appendixes:

In the proofs of Theorems 1 and 2, the dimensionality of the system is reduced until it is 1, and then a trivial solution to the linear program exists.

#### Appendix 1: Proof of Theorem 1

Before proceeding with a proof, recall the definition of the slack variable  $\Psi^K$  given by (2.1):

$$\Psi^K = \sum_{S_i \in S^K} \mu_i - \sum_{A_L \in A^K} \lambda_L$$

The theorem assumes that the slack for each  $K \subset \{1, \dots, n\}$  is strictly greater than zero.

We will be using sets of the form  $\zeta_W^V$ , where  $W, V \subset \{1, \dots, n\}; W \neq \emptyset$ .  $\zeta_W^V$  is defined by

$$\zeta_W^V = \{K \subset \{1, \dots, n\} | V \subset K, W \not\subset K\}$$

Also, define  $\zeta_\emptyset^V$  by

$$\zeta_\emptyset^V = \{K \subset \{1, \dots, n\} | V \subset K\}$$

For example, if  $n = 3$ , then  $\zeta_{\{3\}}^{\{1\}} = \{\{1\}, \{1, 2\}\}; \zeta_\emptyset^{\{1, 2\}} = \{\{1, 2\}, \{1, 2, 3\}\}$ .

We will be interested in the slack quantity  $\Psi_W^V$ , where

$$\Psi_W^V = \min_{K \in \zeta_W^V} \Psi^K$$

Note that  $\Psi_W^V$  is strictly greater than zero since the minimum over a finite set is achieved.

Recall that there is only one class of packets for Theorem 1, so the superscript  $j$  is dropped from all notation, e.g.  $\lambda_K^1 \doteq \lambda_K$ .

Two lemmas will be required in the proof of Theorem 1:

### Lemma 1.1

Given  $D, B \subset \{1, \dots, n\}$ ,

$$\begin{aligned} \sum_{A_K \in A^{D \cup B}} \lambda_K &= \sum_{A_K \in A^D} \lambda_K + \sum_{A_K \in A^B} \lambda_K + \\ &\quad \sum_{A_K \in [A^{D \cup B} - A^D - A^B]} \lambda_K - \sum_{A_K \in A^{D \cap B}} \lambda_K \end{aligned}$$

Proof:

$$A^{D \cup B} = A^D \cup A^B \cup [A^{D \cup B} - A^D - A^B]$$

$$A^D \cap [A^{D \cup B} - A^D - A^B] = A^B \cap [A^{D \cup B} - A^D - A^B] = \emptyset$$

$$A^D \cap A^B = \{A_K | K \subset D\} \cap \{A_K | K \subset B\} = \{A_K | K \subset D \cap B\}$$

The result follows.

### Lemma 1.2

Given  $D, B \subset \{1, \dots, n\}$ ,  $G \subset D \cup B$ ,  $G \not\subset D$ ,  $G \not\subset B$ , and the assumptions of Theorem 1, we have

$$\Psi^D + \Psi^B > \lambda_G$$

Proof:

We know that

$$\Psi^{D \cup B} = \sum_{S_i \in S^{D \cup B}} \mu_i - \sum_{A_K \in A^{D \cup B}} \lambda_K$$

Let us examine the first and second terms of the right side of the above equation. For the first term we have:

$$\sum_{S_i \in S^{D \cup B}} \mu_i = \sum_{S_i \in S^D} \mu_i + \sum_{S_i \in S^B} \mu_i - \sum_{S_i \in S^{D \cap B}} \mu_i$$

The second term (the subtracted term) is:

$$\sum_{A_K \in A^{D \cup B}} \lambda_K$$

which, by lemma 1.1 equals

$$\sum_{A_K \in A^D} \lambda_K + \sum_{A_K \in A^B} \lambda_K + \sum_{A_K \in [A^{D \cup B} - A^D - A^B]} \lambda_K - \sum_{A_K \in A^{D \cap B}} \lambda_K$$

Using the definition of  $\Psi^K$ , we have

$$\Psi^{D \cup B} = \Psi^D + \Psi^B - \Psi^{D \cap B} - \sum_{A_K \in [A^{D \cup B} - A^D - A^B]} \lambda_K <$$

$$\Psi^D + \Psi^B - \sum_{A_K \in [A^{D \cup B} - A^D - A^B]} \lambda_K \leq$$

$$\Psi^D + \Psi^B - \lambda_G$$

Using the fact that  $\Psi^{D \cup B} > 0$ , we have the result.

**Proof (Theorem 1):**

For every  $K \subset \{1, \dots, n\}$ , we will use Bernoulli splitting of the arrival process  $A_K$ , so that when an arrival from the process  $A_K$  occurs, it will be routed to queue  $q_i$  with probability  $p_K^i$ , where  $p_K^i = 0$  for  $i \notin K$ , and  $\sum_{i \in K} p_K^i = 1$ . For  $W \subset K$ , define  $p_K^W = \sum_{i \in W} p_K^i$ . Therefore, we have  $p_K^K = 1$ .

We now define a method to construct  $\{p_K^i\}$  for each arrival process  $A_K$ . For  $K = \{k\}$ , we have  $p_K^k = 1$  and we are done (Note that  $A_{\{k\}}$  is an uncontrolled arrival process – it is routed to  $q_{\{k\}}$ ). Suppose that  $|K| = l > 1$ . Let  $K = \{a_1, \dots, a_l\}$ . We now define a recursive method to determine  $p_K^{a_i}; 1 \leq i \leq l$ .

We know that  $\exists B, D \neq \emptyset$  such that  $K = D \cup B$  and  $D \cap B = \emptyset$ . Also, we know that the minimum which defines  $\Psi_D^B$  is achieved. Let us say that  $U \subset \zeta_D^B$  achieves the minimum. We also know that  $\exists V \subset \zeta_B^D$  which achieves the minimum in  $\Psi_B^D$ . From lemma 1.2 we have that  $\Psi^V + \Psi^U > \lambda_K$ . Therefore,

$$\Psi_D^B + \Psi_B^D > \lambda_K$$

Therefore,  $\exists b, d > 0$  such that  $b + d = \lambda_K$  and

$$b < \Psi_D^B; d < \Psi_B^D$$

We claim that if  $\lambda_B$  is increased by  $b$ ,  $\lambda_D$  is increased by  $d$ , and  $\lambda_K$  is set to zero, the equation

$$\sum_{S_i \in S^W} \mu_i > \sum_{A_{a_1, \dots, a_l} \in A^W; 1 \leq l \leq |W|} \lambda_{a_1, \dots, a_l}$$

still holds  $\forall W \subset \{1, \dots, n\}$ .

Proof of claim: Examination of the above equation (which is equivalent to the condition  $\Psi^W > 0 \forall W \subset \{1, \dots, n\}$ ), reveals that only the value of

$$\sum_{A_{a_1, \dots, a_l} \in A^W; 1 \leq l \leq |W|} \lambda_{a_1, \dots, a_l}$$

is affected by the change and that it is only affected for  $W \in [\zeta_D^B \cup \zeta_B^D]$ . (The value is unaffected for  $W \in \zeta_\emptyset^K$  since  $b$  and  $d$  are subtracted, but  $\lambda_K$  is added to  $\Psi^W$ , and it is totally unaffected for  $W \in [\zeta_B^\emptyset \cap \zeta_D^\emptyset]$ ) Recall that  $b$  and  $d$  were chosen so that

$$b < \Psi_D^B; d < \Psi_B^D$$

Therefore, the value of  $\Psi^W$  is reduced, but still greater than zero for  $W \in \zeta_D^B \cup \zeta_B^D$ . Since

$$\{W \subset \{1, \dots, n\}\} = [\zeta_D^B \cup \zeta_B^D] \cup \zeta_\emptyset^K \cup [\zeta_B^\emptyset \cap \zeta_D^\emptyset]$$

The claim is proved.

We have just seen that if the  $A_K$  is Bernoulli split by defining  $p_K^B = \frac{b}{\lambda_K}$  and  $p_K^D = \frac{d}{\lambda_K}$ , and if an  $A_K$  arrival split to  $B$  (respectively  $D$ ) is treated exactly like an  $A_B$  arrival (respectively an  $A_D$  arrival), then the  $\Psi$ s are all still strictly greater than zero.

Let us begin with the system given in the assumption of theorem 1. First, we Bernoulli split the  $A_{\{1, \dots, n\}}$  arrival process into two pieces, and record what the two subsets and their Bernoulli probabilities are. We now have a new system satisfying the assumption of theorem 1. Next, split each of the arrival processes  $A_K$ , where  $|K| = n - 1$ . Record their Bernoulli splitting probabilities as well as the subsets they were split into and update the values of the  $\lambda$ s accordingly. Then split each of the arrival processes with  $|K| = n - 2$ . Proceed until all of the controlled arrival processes have zero arrival rate. By construction, it holds that

$$\lambda_i < \mu_i \forall i \in \{1, \dots, n\}$$

so the resulting system of  $M/M/1$  queues is trivially stable. The Bernoulli splitting probability  $p_K^i$  can be found by taking the Bernoulli probability of the subset containing  $i$  that  $K$  was split into, multiplying it by the splitting probability of the subsequent subset that contains  $i$  which that subset was split into, and proceeding until you are split into  $\{i\}$  itself.

We have specified a method to construct (given the  $\lambda$ s and the  $\mu$ s) an open loop Bernoulli splitting routing policy which stabilizes the “controlled arrival process” routing problem.

## Appendix 2: Proof of Theorem 2

Rather than prove Theorem 2 as stated, we prove a version in discrete time stated as Theorem 2a. Before stating Theorem 2a, we introduce the discrete time controlled service process system with a single server which may serve any one of a randomly selected subset of the queues at each time instant. It can be shown through uniformization [5] that the Theorem 2a is equivalent to Theorem 2. Thus, not only is a discrete-time system equivalent to the continuous time system of Theorem 2, but the discrete time system has a single server with random, time-varying connectivity rather than multiple servers. The discrete time controlled service process system is equivalent to a single-server system introduced by Tassiulas and Ephremides [1]. It is important to note that while the controlled service process problem can be reduced to the case of a single server, the controlled arrival process problem cannot, even through uniformization.

The system which Tassiulas and Ephremides address in [1] is a discrete-time system with a single server and  $n$  queues. At each time instant, there are  $A_i^t$  arrivals at queue  $q_i$  where  $E[A_i^t] = a_i$ , and  $A_i^t$  is bounded  $\forall t$ . The number of arrivals to queue  $q_i$  at time  $t_1$  is independent of and identically distributed to the number of arrivals to queue  $q_i$  at time  $t_2$ . At each time  $t$ , a subset  $W_t \subset \{1, \dots, n\}$  of the queues is connected to the server.  $W_t$  is distributed as follows:

$$P[W_t = K] = P_K^t, K \subset \{1, \dots, n\}$$

Tassiulas and Ephremides define a connectivity variable  $C_i^t$  such that  $C_i^t = 1$  if  $i \in W_t$  and  $C_i^t = 0$  otherwise. They require that  $C_i^{t_1}$  and  $C_i^{t_2}$  are independent and identically distributed. This is ensured by the stronger requirement that  $P_K^t = P_K \forall t$ . However, in their remark 4 (p. 472), they generalize their result to the case of dependent connectivity variables at time  $t$  (Their derivation assumed the independence of  $C_i^t$  and  $C_j^t; i \neq j$ .) Therefore, this stronger requirement is appropriate and parsimonious. At each time  $t$ , the server decides which connected queue to serve based on the current queue lengths, the history of the queue lengths and the history of past decisions. If queue  $q_i$  is chosen for service at time  $t$ , the service is completed successfully if the random variable  $M_i^t = 1$ , and it is unsuccessful if  $M_i^t = 0$ . It is also required that  $M_i^{t_1}$  and  $M_i^{t_2}$  are independent and identically distributed, and the expected value of  $M_i$  is denoted by  $m_i$ . Finally, the arrival, connectivity, and service completion random variables are independent.

In [1], it was determined that the system is stabilizable iff

$$\sum_{i \in K} \frac{a_i}{m_i} < \sum_{W \in S^K} P_W; \forall K \subset \{1, \dots, n\}$$

(see their eqn 3.2, p. 468 and remark 4, p. 472)

where  $S^K$  is defined by

$$S^K = \{W \subset \{1, \dots, n\} | W \cap K \neq \emptyset\}$$

Remark:  $S^K$  refers to a set of subsets of the  $n$  queues which may be connected to the server at a particular time.

We can now state Theorem 2a:

### Theorem 2a

The discrete time controlled service process problem is stabilizable iff

$$\sum_{i \in K} \frac{a_i}{m_i} < \sum_{W \in S^K} P_W; \forall K \subset \{1, \dots, n\}$$

Furthermore, if the system is stabilizable, then there exists an open loop stabilizing policy that can be determined directly through a recursive algorithm analogous to Algorithm 2.

Before we proceed to the proof, we need to introduce some notation and prove two lemmas. First, a brief explanation:

Similarly to the proof of Theorem 1, we shall construct a stabilizing policy using Bernoulli splitting. Instead of splitting at the arrival process  $A_K$ , we now split at the service process  $W_t$ . For each  $K \in \{1, \dots, n\}$ , we decide on the probabilities  $p_K^i$  of serving queue  $q_i$  given that the set of connected queues at time  $t$  is  $W_t = K$ .

Similarly to Theorem 1, we introduce some notation. Define  $\Psi^k$ , the “slack” available between the connectivity rate and arrival of work rate for subset  $K$  by

$$\Psi^K = \sum_{W \in S^K} P_W - \sum_{i \in K} \frac{a_i}{m_i} > 0$$

Remark: The above definition is the same as (3.1), where now we use  $P_W$  in place of  $\mu_W$  and  $\frac{a_i}{m_i}$  in place of  $\lambda_i$ .

**Lemma 2.1**

Given  $D, B \subset \{1, \dots, n\}$ ,

$$\sum_{K \in S^{D \cup B}} P_K = \sum_{K \in S^D} P_K + \sum_{K \in S^B} P_K - \sum_{K \in S^{D \cap B}} P_K - \sum_{K \in [(S^D \cap S^B) - S^{D \cap B}]} P_K$$

Proof:

$$S^{D \cup B} = \{K \subset \{1, \dots, n\} | K \cap \{D \cup B\} \neq \emptyset\} =$$

$$\{K \subset \{1, \dots, n\} | \{K \cap D\} \cup \{K \cap B\} \neq \emptyset\} =$$

$$\{K \subset \{1, \dots, n\} | K \cap D \neq \emptyset\} \cup \{K \subset \{1, \dots, n\} | K \cap B \neq \emptyset\} =$$

$$S^D \cup S^B$$

Also,

$$S^D \cap S^B = S^{D \cap B} \cup [(S^D \cap S^B) - S^{D \cap B}]$$

and

$$S^{D \cap B} \cap [(S^D \cap S^B) - S^{D \cap B}] = \emptyset$$

The result follows.

**Lemma 2.2**

Given  $D, B, G \subset \{1, \dots, n\}$ , such that  $G \cap D \neq \emptyset$ ,  $G \cap B \neq \emptyset$ ,  $G \cap D \cap B = \emptyset$  and given that the assumptions of Theorem 2 hold, we have

$$\Psi^D + \Psi^B > P_G$$

Proof:

We know that

$$\Psi^{D \cup B} = \sum_{K \in S^{D \cup B}} P_K - \sum_{i \in S^{D \cup B}} \frac{a_i}{m_i}$$

By lemma 2.1, the first term on the right side of the above equation can be written as

$$\sum_{K \in S^{D \cup B}} P_K = \sum_{K \in S^D} P_K + \sum_{K \in S^B} P_K - \sum_{K \in S^{D \cap B}} P_K - \sum_{K \in [(S^D \cap S^B) - S^{D \cap B}]} P_K$$

The second (subtracted) term can be written as

$$\sum_{i \in S^{D \cup B}} \frac{a_i}{m_i} = \sum_{i \in S^D} \frac{a_i}{m_i} + \sum_{i \in S^B} \frac{a_i}{m_i} - \sum_{i \in S^{D \cap B}} \frac{a_i}{m_i}$$

So, by the definition of  $\Psi^K$ , we can write

$$\Psi^{D \cup B} = \Psi^D + \Psi^B - \Psi^{D \cap B} - \sum_{K \in [(S^D \cap S^B) - S^{D \cap B}]} P_K$$

So we have that

$$0 < \Psi^{D \cup B} + \Psi^{D \cap B} = \Psi^D + \Psi^B - \sum_{K \in [(S^D \cap S^B) - S^{D \cap B}]} P_K \leq$$

$$\Psi^D + \Psi^B - P_G$$

We now introduce more new notation. Please note that  $\Psi^K$ ,  $\zeta_D^B$ , and  $\Psi_D^B$  are defined differently here than they were in the proof of Theorem 1.

Define  $\zeta_B^D$  as

$$\zeta_B^D = \{K \subset \{1, \dots, n\} | D \cap K \neq \emptyset, B \cap K = \emptyset\}$$

Furthermore, define  $\zeta_B^\emptyset$  as

$$\zeta_B^\emptyset = \{K \subset \{1, \dots, n\} | B \cap K = \emptyset\}$$

And define  $\Psi_B^D$  as

$$\Psi_B^D = \min_{K \in \zeta_B^D} \Psi^K$$

We are now ready to proceed with the proof of Theorem 2.

Proof:

We now describe how to split  $P_G$  into two pieces so that the assumptions of Theorem 2a still hold. The rest of the proof (i.e. constructing the branching leading down to the individual  $P_{\{i\}}$ s, and defining the Bernoulli splitting probabilities as the product of the splitting probabilities going down the branching) is identical to the proof of Theorem 1, and is not repeated here.

Let  $G \subset \{1, \dots, n\}$ ,  $|G| \geq 2$ . Then  $\exists B, D \subset \{1, \dots, n\}$  such that  $B, D \neq \emptyset$ ,  $B \cap D = \emptyset$ , and  $B \cup D = G$ . Let  $U \subset \{1, \dots, n\}$  achieve the minimum over  $K \in \zeta_B^D$  in  $\Psi_B^D$ . Also, let  $V$  achieve the minimum over  $K \in \zeta_D^B$  in  $\Psi_D^B$ . We know by lemma 2.2 that  $\Psi^U + \Psi^V > P_G$ . Choose  $b, d > 0$  such that  $b < \Psi^U = \Psi_B^D$ ,  $d < \Psi^V = \Psi_D^B$ , and  $b + d = P_G$ . We claim that if  $P_G$  is set to zero,  $P_B$  is increased by  $b$ , and  $P_D$  is increased by  $d$ , then the assumptions of Theorem 2 still hold.

Proof of claim:

We must verify that we still have that  $\Psi^K > 0$ ,  $\forall K \subset \{1, \dots, n\}$ . If  $K \in \zeta_{B \cup D}^\emptyset$ , then  $P_G$ ,  $P_B$ , and  $P_D$  have no effect on the value of  $\Psi^K$ , so it is still greater than zero. Similarly, if  $K \in [\zeta_\emptyset^D \cap \zeta_\emptyset^B]$ , then the value of  $\Psi^K$  remains unchanged because  $b$  and  $d$  are added to it, while  $P_G$  is subtracted from it. If  $K \in \zeta_D^B$ , then  $P_G$  is subtracted from  $\Psi^K$  while  $b$  is added to it. So in effect,  $d$  is subtracted from it. But  $d < \Psi_D^B < \Psi^K$ , so  $\Psi^K$  is still larger than zero. By an analogous argument (switch the  $bs$  and  $ds$ ), if  $K \in \zeta_B^D$ , then  $\Psi^K$ , while reduced is still greater than zero. Since

$$\{K \subset \{1, \dots, n\}\} = \zeta_B^D \cup \zeta_D^B \cup \zeta_{B \cup D}^\emptyset \cup [\zeta_\emptyset^D \cap \zeta_\emptyset^B]$$

the claim is proved.

If we reduce down to the  $P_{\{i\}}$ s as in the proof of Theorem 1, then we have by construction that

$$\frac{a_i}{m_i} < P_{\{i\}} \quad \forall i \in \{1, \dots, n\}$$

Since we know that

$$E[x_i^{t+1} - x_i^t] = E[A_i^t - M_i^t P_{\{i\}}^t] =$$

$$E[A_i^t] - E[M_i^t] E[P_{\{i\}}^t] = a_i - m_i P_{\{i\}} < 0$$

and we know that the Markov chain defined under this Bernoulli splitting policy is irreducible and that  $x_i^{t+1} - x_i^t$  is bounded since  $A_i^t$  is bounded, we have by Foster's theorem on ergodicity of a Markov chain ([6]) that the system is stable.